

FIG. 1

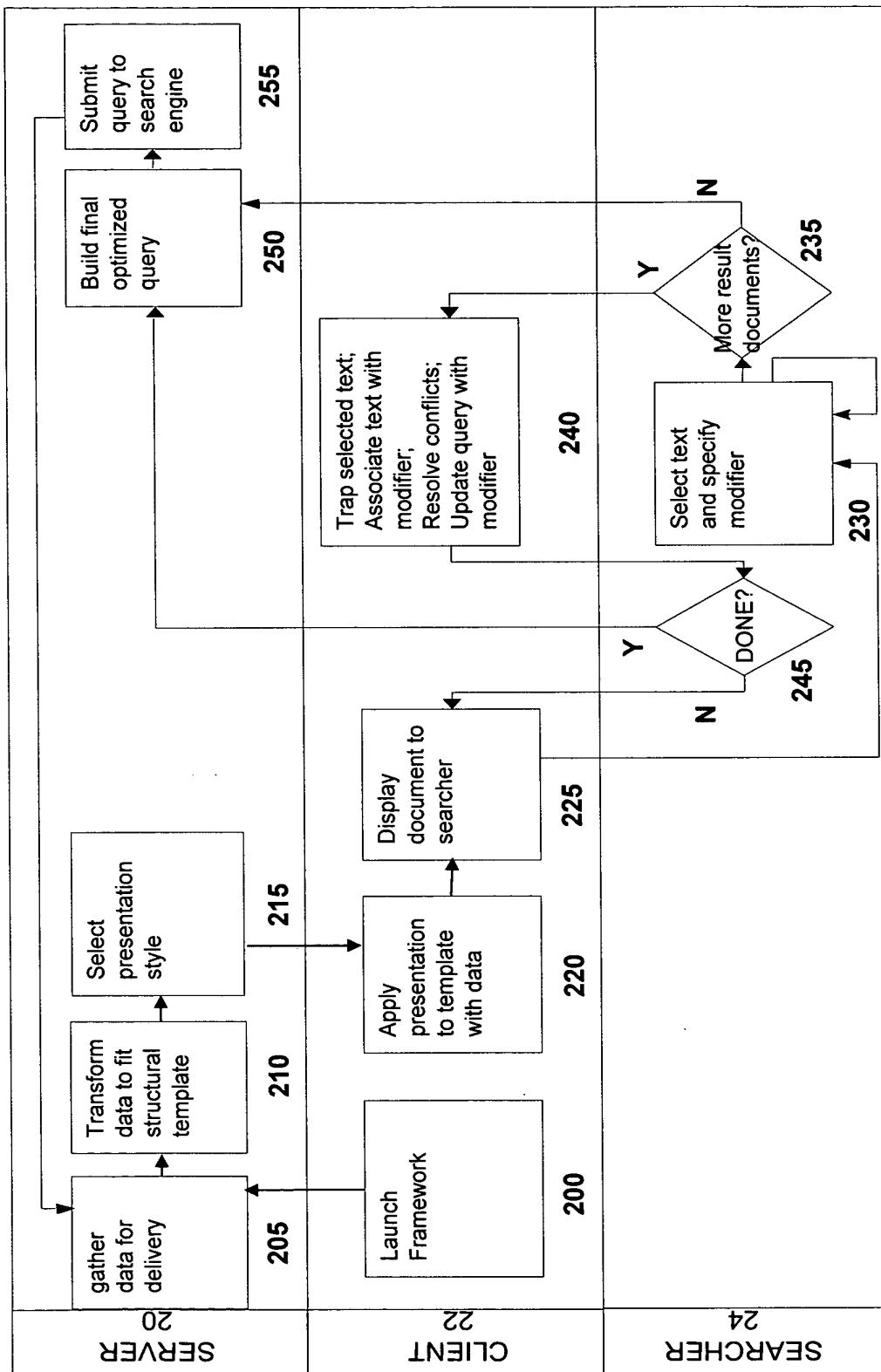


FIG. 2A

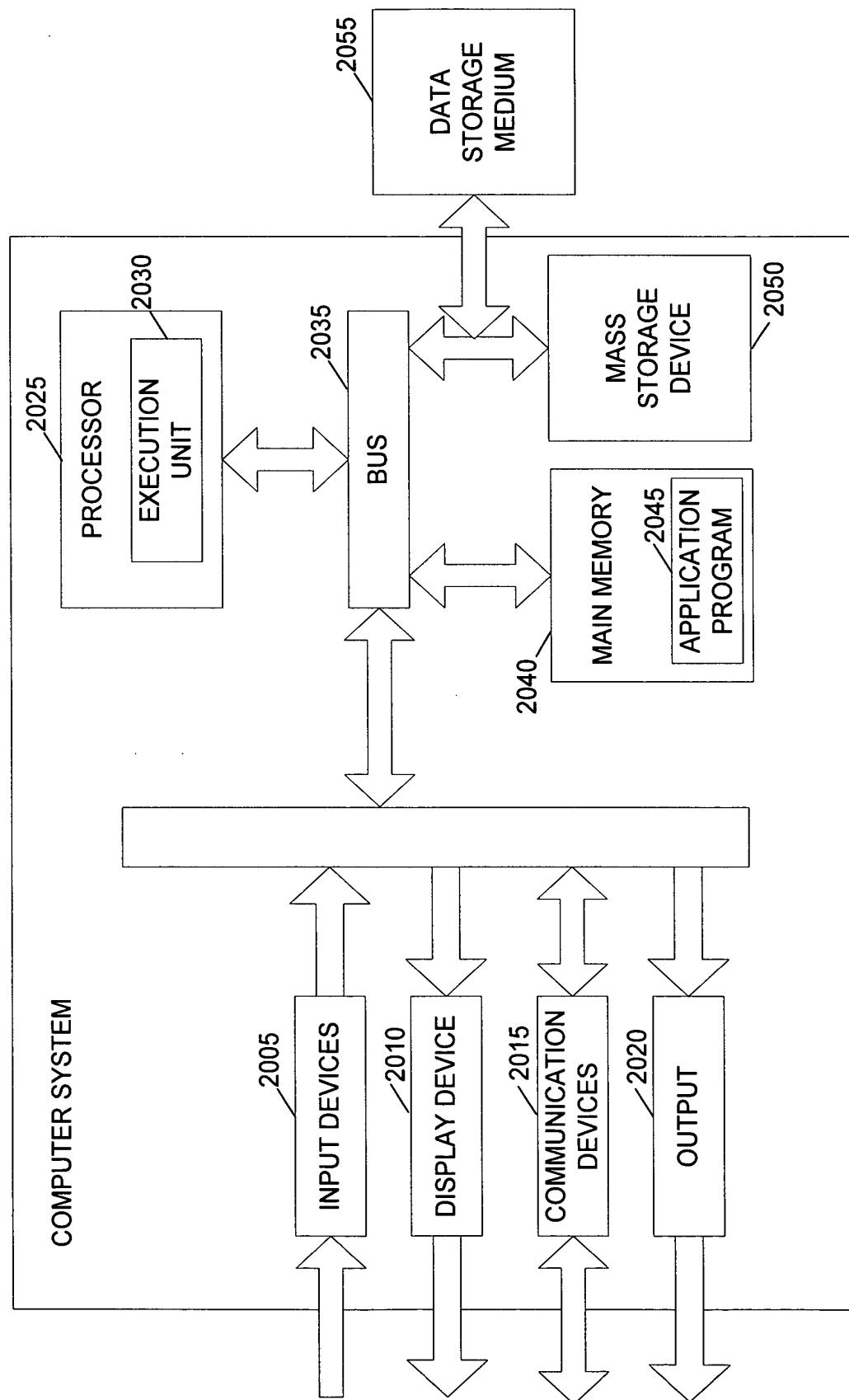


FIG. 2B

```
300  if (document.structureType == XML) {
301      XMLGrammar = document.retrievedDTD();
302      if (XMLGrammar.DTDType == Physical) {
303          XMLElementWeights = loadElementWeights(elementWeightFile);
304      }
305      else {
306          structuralElements = document.buildStructure(XMLGrammar);
307      }
308  }
309  else {
310      if (document.wellFormed) {
311          pseudoStructuralElements = loadStructureTemplate(structureTemplateFile);
312      }
313      else {
314          keywordList = NULL;
315      }
316  }
317  if (structuralElements) return (TypeIV, structuralElements);
318  if (XMLElementWeights) return (TypeIII, XMLElementWeights);
319  if (pseudoStructuralElements) return (TypeII, pseudoStructuralElements);
320  return (TypeI, keywordList);
```

FIG. 3

5/8

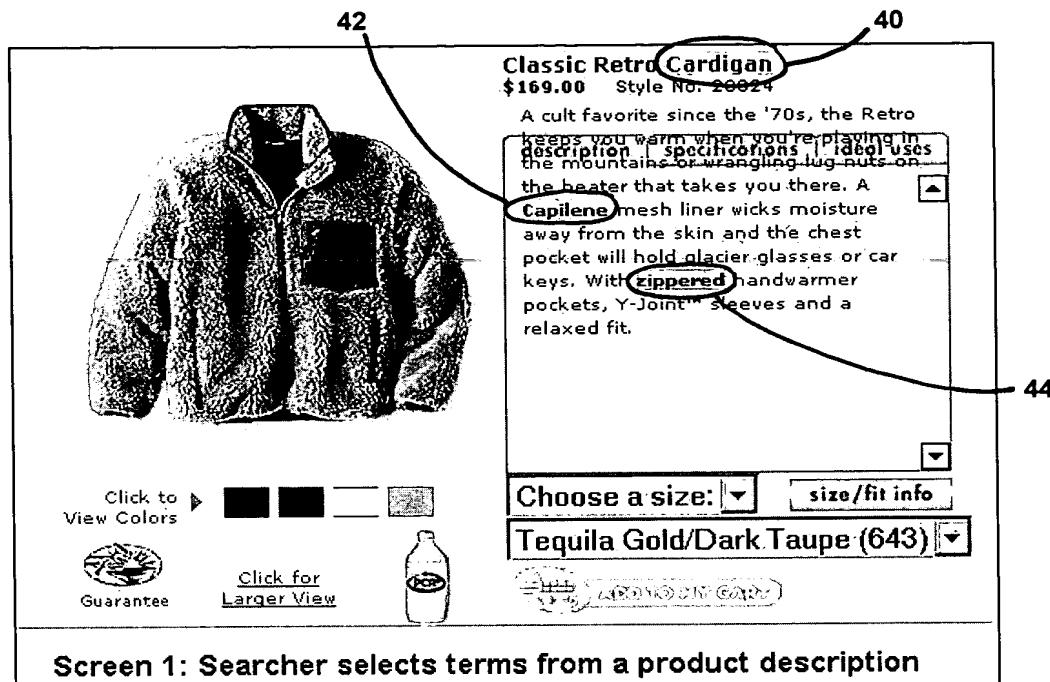


FIG. 4A

```
<?xml version="1.0"?>
<!DOCTYPE Recipe SYSTEM "product.dtd">

<PRODUCT_PAGE>

  <PRODUCT>
    <ITEM_ATTRIBUTES>
      <NAME>Classic Retro Cardigan</NAME>
      <STYLE_NO>23024</STYLE_NO>
      <PRICE>$169.00</PRICE>
    </ITEM_ATTRIBUTES>

    <ITEM_DETAILS>
      A cult favorite since the '70s, the Retro keeps you warm when you're playing in the mountains or wrangling lug nuts on the beater that takes you there. A Capilene mesh liner wicks moisture away from the skin and the chest pocket will hold glacier glasses or car keys. With zippered handwarmer pockets, Y-Joint™ sleeves and a relaxed fit.
    </ITEM_DETAILS>
  </PRODUCT>
</PRODUCT_PAGE>
```

FIG. 4B

```
FOR $item IN
  document("data/productCatalog.xml")//PRODUCT
WHERE
  CONTAINS($item/ITEM_ATTRIBUTES/NAME, "Cardigan") OR
  CONTAINS($item/ITEM_DETAILS, "Capilene") AND
  NOT(CONTAINS($item/ITEM_DETAILS, "zippered"))
RETURN
<RESULT_LIST>
  $item
</RESULT_LIST>
```

FIG. 4C

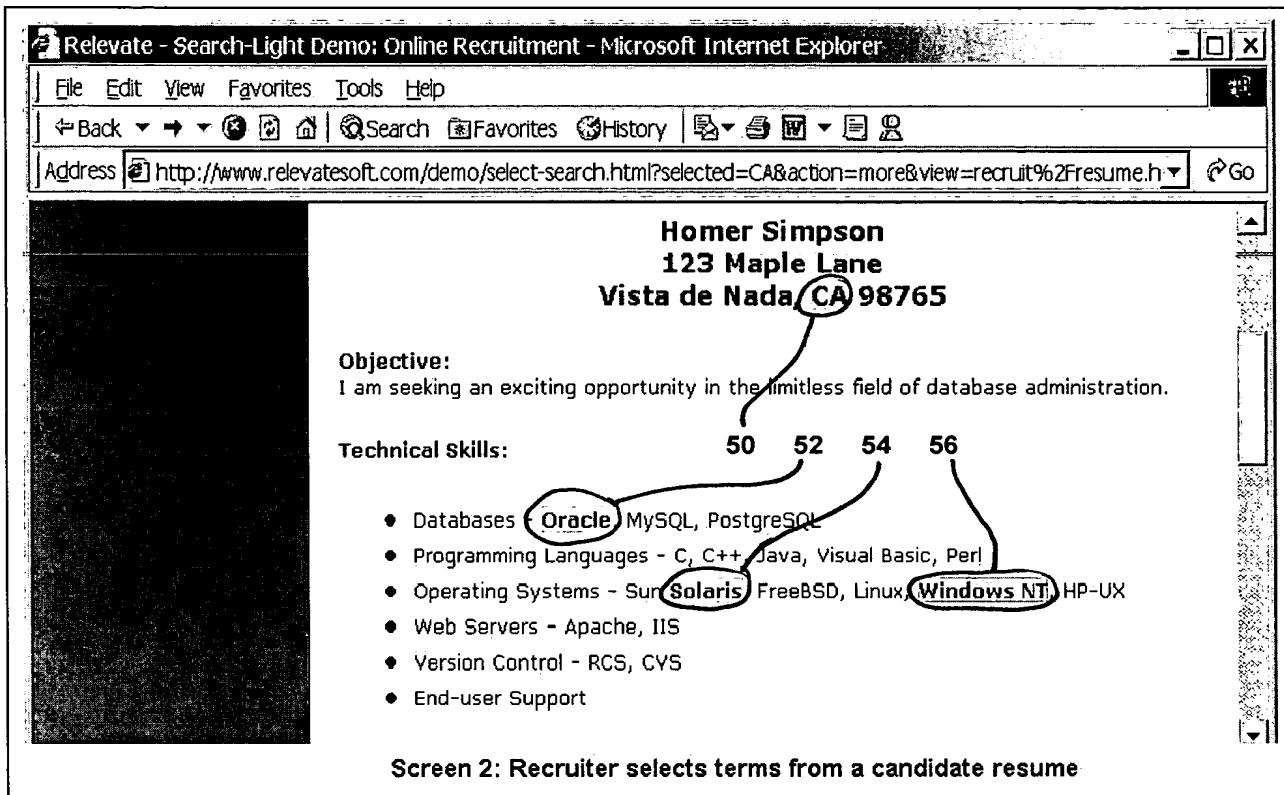


FIG. 5A

```
<b>Homer Simpson<br>
123 Maple Lane<br>
Vista de Nada, CA 98765<br>
</b>
:
<ul>
<li>Databases - Oracle, MySQL, PostgreSQL</li>
<li>Programming Languages - C, C++, Java, Visual Basic, Perl</li>
<li>Operating Systems - Sun Solaris, FreeBSD, Linux, Windows NT, HP-
UX</li>
:
</ul>
```

FIG. 5B

```
DEFINE FUNCTION getScore(element $item) RETURNS INTEGER* {
    <compute score based on weights>
    RETURN $score
}
FOR $item IN input()//resume
LET $MLTList := ("CA", "Oracle", "Solaris")
LET $LLTList := ("Windows NT")
WHERE
    SOME $b IN $item/B AND SOME $MLTTerm IN $MLTList AND SOME $LLTTerm
    in $LLTList SATISFIES
        (CONTAINS ($b/text(), $MLTTerm/text())) AND
        NOT(CONTAINS ($b/text(), $LLTTerm/text())))
    OR SOME $l IN $item/LI AND SOME $MLTTerm IN $MLTList AND SOME
    $LLTTerm in $LLTList SATISFIES
        (CONTAINS ($l/text(), $MLTTerm/text())) AND
        NOT(CONTAINS ($l/text(), $LLTTerm/text())))
RETURN
<RESULT_LIST>
    <RESULT>$item</RESULT>
    <SCORE>getScore($item)</SCORE>
</RESULT_LIST>
```

FIG. 5C